



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/994,483	11/26/2001	Michael A. Eidson	10559-515001/ P12419	7245
20985	7590	09/21/2006	EXAMINER	
FISH & RICHARDSON, PC P.O. BOX 1022 MINNEAPOLIS, MN 55440-1022			PETRANEK, JACOB ANDREW	
			ART UNIT	PAPER NUMBER
			2183	

DATE MAILED: 09/21/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b> 09/994,483	<b>Applicant(s)</b> EIDSON ET AL.	
	<b>Examiner</b> Jacob Petranek	<b>Art Unit</b> 2183	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 24 July 2006.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-14,23-32 and 39-42 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-14,23-32 and 39-42 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

Art Unit: 2183

### **DETAILED ACTION**

1. Claims 1-14, 23-32, and 39-42 have been examined.
2. The office acknowledges the following papers:

Claims and arguments filed 7/24/2006.

### ***Withdrawn objections***

3. The specification objection has been withdrawn.
4. The 35 USC § 102 claim rejections for claims 1, 7, 15, 17, 19, 21, 23, and 25 have been withdrawn due to amendments and cancelled claims.

### ***New Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1-3, 6, 27, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fowler (U.S. 4,502,116), in view of Takase (U.S. 6,289,473).

7. As per claim 1:

Fowler disclosed a method comprising:

In response to a first signal (hardware or software induced interrupt) indicating that a processor has encountered the breakpoint (test utility system, not explicitly shown

Art Unit: 2183

in drawings), suspending execution of the first peripheral and saving the state of the first peripheral (target subsystem is the first peripheral): (Column 1, lines 26-31, Column 2, lines 59 to column 3, line 12, Steps 1-4 in Column 9, line 42-65)

Continuing execution of the breakpoint by the processor and the first peripheral in response to receiving a second signal indicating that the state of the first peripheral has been saved: (Step 5 in Column 9, line 66 to Column 10, line 22 indicates the peripheral's state has been saved, and afterwards in step 6, the utility, in response to the saved state being completed, continues execution by carrying out the appropriate action, which as stated in column 2, lines 6-25 step 3 includes means to display and modify registers and column 3, lines 7-12 and column 1, lines 26-31. Since step 6 is the next sequential step in the synchronized process in Column 9, line 42-65 and includes modifying registers, it is inherent the saving of the state of the peripheral must be complete in step 5 before step 6 is started); and

Restoring the saved state of the first peripheral in response to a third signal indicating that execution of the breakpoint by the processor has been completed: (The Test Utility Processor initiates a return to the application program in step 7 (shown in column 9, lines 42-65) via "an assert resume" portion of its software which also indicates the breakpoint has been completed. In the next step (step 8), after the initiation in step 7, the state of the subsystem is restored. (Column 2, lines 66-68 and Column 9, lines 42-65).

While Fowler teaches that the context/state of the target processor is saved after a breakpoint and before each individual processor is debugged/monitored, Fowler does

Art Unit: 2183

not teach receiving an indication that a first peripheral is to participate in an execution of a breakpoint whereas a second peripheral is not to participate; that the processor which initiates the breakpoint waits for an explicit signal from the target processor which indicates that the target processor has completed saving the state; and to continue with the breakpoint handling.

Takase teaches receiving an indication that a first peripheral is to participate in an execution of a breakpoint whereas a second peripheral is not to participate (Takase: Column 9 lines 11-26)(Fowler: Column 9 lines 40-50)(The combination of Takase and Fowler results in multiple peripherals within the processor. Fowler states that the source of the interrupt is known and that the source is involved in the process of executing a breakpoint. Thus, any other peripherals don't participate in the execution of the breakpoint.)

A module that receives a signal that a breakpoint has occurred, (col. 8, line 48 to col. 9, line 19 describe the breakpoint and the signal that indicates a breakpoint has occurred). The state of the module is first saved (col. 9, lines 23-26), and then it outputs a stop notice signal, which indicates the module is stopped and the state has been kept, and then the debugger continues with the execution of the breakpoint. (Column 9, lines 20-42.) One of ordinary skill in the art would have recognized that using explicit signals, as taught in Takase, ensures the system is in a known, stable, and paused state before entering into the debugging, which, in turn, would cause accurate debugging of the system and prevent altering of data prior to it being saved.

Art Unit: 2183

It would have been obvious to one of ordinary skill in the art to have a signal to indicate that the state has finished being saved. Fowler teaches a synchronized breakpoint system and debugging each of the plurality of processors during a breakpoint. One of ordinary skill in the art would have recognized that ensuring the system is in a known, stable, and paused state before entering into the debugging operations would cause accurate debugging of the system and prevent altering of data prior to it being saved. This can be ensured using an explicit signal which indicates the processors are in a stopped state and their state is saved, such as that taught by Takase.

8. As per claim 2:

Fowler and Takase disclosed the method of claim 1 comprising resuming normal execution of the processor in response to a signal indicating that the saved state has been restored. (Step 10, which follows the restoring of the state of the peripheral in step 8, includes releasing a pause-out condition. Following the release of the pause-out condition, the utility processor returns to normal operation. Normally the utility processor has interrupts enabled, but they are disabled for the breakpoint handling, they are re-enabled in step 10, and hence, the utility processor resumes normal operation. Column 9, line 42-65).

9. As per claim 3:

Fowler and Takase disclosed the method of claim 1 comprising resuming normal execution of the first peripheral and the second peripheral in response to a signal indicating that the processor has resumed normal execution: (Normally the utility

Art Unit: 2183

processor has interrupts enabled, but they are disabled for the breakpoint handling, they are re-enabled in step 10, and hence, the utility processor resumes normal operation.

After the normal operation of the processor has resumed, indicated by the pause-in signal going inactive, the target subsystem resumes normal execution. Steps 10-15 in Column 9, line 42-65, also column 9, lines 23-31)(Takase: Column 9 lines 11-26)(The other peripherals are restarted with a previous state upon the breakpoint completion.).

10. As per claim 6:

Fowler and Takase disclosed the method of claim 1 comprising triggering the breakpoint in response to a condition associated with occurrence of an instruction being executed by the processor. (Fowler teaches that the test utility system provides a breakpoint feature in order to halt/pause execution of a target subsystem and to proceed with testing and/or modifying the subsystem. Breakpoint is defined "to be set when both a point in the program and an event that will cause the suspension of execution at that point are defined" (The Authoritative Dictionary of IEEE Standards Terms, 7<sup>th</sup> Ed.) It is therefore inherent that when the test utility system implements the breakpoint feature in order to halt execution and test, it is when both a point in the program (occurrence of an instruction) and an event (condition) associated with the instruction has occurred.

11. As per claim 27:

Fowler and Takase disclosed the method of claim 1, wherein saving the state of the first peripheral comprises saving the state of an input/output device. (Fowler teaches that the peripheral (target subsystem) is a processor. [Col. 2, lines 3-9] The

Art Unit: 2183

processor has inputs and outputs [Figs. 1-4], and therefore is an input/output device, and its state is saved. [Column 1, lines 26-31, Column 2, lines 59 to column 3, line 12, Steps 1-4 in Column 9, line 42-65])

12. As per claim 29:

Fowler and Takase disclosed the method of claim 1.

Fowler and Takase failed to teach wherein suspending execution of the first peripheral comprises suspending execution of a peripheral that is monolithically fabricated on a same chip as the processor.

However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to make the multiple processors be integrated onto one monolithically fabricated chip since it has been held "that the use of a one piece construction instead of the structure disclosed in [the prior art] would be merely a matter of obvious engineering choice." [In re Larson, 340 F.2d 965, 968, 144 USPQ 347, 349 (CCPA 1965)]

Furthermore, Examiner takes Official Notice that putting hardware on a single, monolithically fabricated chip instead of separate chips for different pieces of hardware is well known and conventional in the art.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the peripherals of Fowler, in view of Takase, on a single, monolithically fabricated chip, since Examiner takes Official Notice this is a well known and conventional technique in the art.



Art Unit: 2183

13. Claims 4-5, 7-14, 23-26, 30, 32, 39-40, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fowler (U.S. 4,502,116), in view of Takase (U.S. 6,289,473), further in view of Yee et al. (U.S. 6,598,178).

14. As per claim 4:

Fowler and Takase disclosed the method of claim 1.

Fowler and Takase failed to teach wherein receiving the indication comprises receiving a setting of a register that determines whether generation of the second signal is to be based on the state of the first peripheral.

However, Yee disclosed wherein receiving the indication comprises receiving a setting of a register that determines whether generation of the second signal is to be based on the state of the first peripheral (Yee: Figure 2 element 202, column 5 lines 26-53)(The breakpoint control register allows for certain peripherals to receive a breakpoint and disable breakpoints for others. The signals for saving a state, completing the breakpoint, and resuming normal execution are based on the control register.).

The process of debugging systems has become more complicated as the number of devices connected to processors increase. Testing and debugging becomes a problem of not just testing individual components, but many components in combination. The ability to selectively see certain peripherals and not others assists in more accurately evaluating a plurality of devices at once (Yee: Column 2 lines 49-67 continued to column 3 lines 1-3). One of ordinary skill in the art would have been motivated by allowing more efficient debugging techniques to implement control registers to control which peripherals are being debugged at any given time. Thus, it

Art Unit: 2183

would have been obvious to one of ordinary skill in the art at the time of the invention to implement a control register that indicates which peripherals can be debugged at any given time for the advantage of increased efficiency in the debugging process.

15. As per claim 5:

Fowler and Takase disclosed the method of claim 1.

Fowler and Takase failed to teach wherein receiving the indication comprises receiving a setting of a register that determines whether generation of a signal indicating that the saved state has been restored is based on the state of the first peripheral.

However, Yee disclosed wherein receiving the indication comprises receiving a setting of a register that determines whether generation of a signal indicating that the saved state has been restored is based on the state of the first peripheral (Yee: Figure 2 element 202, column 5 lines 26-53)(The breakpoint control register allows for certain peripherals to receive a breakpoint and disable breakpoints for others. The signals for saving a state, completing the breakpoint, and resuming normal execution are based on the control register.).

The process of debugging systems has become more complicated as the number of devices connected to processors increase. Testing and debugging becomes a problem of not just testing individual components, but many components in combination. The ability to selectively see certain peripherals and not others assists in more accurately evaluating a plurality of devices at once (Yee: Column 2 lines 49-67 continued to column 3 lines 1-3). One of ordinary skill in the art would have been motivated by allowing more efficient debugging techniques to implement control

Art Unit: 2183

registers to control which peripherals are being debugged at any given time. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a control register that indicates which peripherals can be debugged at any given time for the advantage of increased efficiency in the debugging process.

16. As per claim 7:

Fowler disclosed a system comprising:

A processor (Test utility system);

A first computer-readable medium storing instructions that, when applied to the processor, cause the processor to: (The Test utility system contains software to run a routine (Column 2, line 59 to column 3, line 17 and column 9, line 14 to column 10, line 22), it is inherent that the instructions in the program are stored on a computer-readable medium or else the computer would not be able to read them)

Generate a first signal indicating execution of a breakpoint: (A breakpoint causes the target subsystem to halt execution, which is done via a pause/resume signal. (Steps 1-4, Column 9, lines 42-65, Column 1, lines 18-31 and column 9, lines 14-39))

Continue execution of the breakpoint after the state of the peripheral has been saved: (Step 5 in Column 9, line 66 to Column 10, line 22 indicates the peripheral's state has been saved, and afterwards in step 6, the utility, in response to the saved state being completed, continues execution by carrying out the appropriate action, which as stated in column 2, lines 6-25 step 3 includes means to display and modify registers and column 3, lines 7-12 and column 1, lines 26-31. Since step 6 is the next sequential step in the synchronized process in Column 9, line 42-65 and includes

Art Unit: 2183

modifying registers, it is inherent the saving of the state of the peripheral must be complete in step 5 before step 6 is started)

And generate a third signal indicating that execution of the breakpoint has been completed: (The Test Utility Processor initiates a return to the application program in step 7 (shown in column 9, lines 42-65) via "an assert resume" portion of its software which also indicates the breakpoint has been completed. In the next step (step 8), after the initiation in step 7, the state of the subsystem is restored. (Column 2, lines 66-68 and Column 9, lines 42-65).

A first peripheral coupled to the processor: (Target subsystem);

A second computer-readable medium storing instructions that, when applied to the first peripheral, cause the first peripheral to:

Suspend execution and save a state of the first peripheral, in response to receiving the first signal: (Step 5 in column 9, lines 42-65; Column 1, lines 26-31, Column 2, lines 59 to column 3, line 12);

Continue execution of the breakpoint in response to receiving the second signal (Step 5 in Column 9, line 66 to Column 10, line 22 indicates the peripheral's state has been saved, and afterwards in step 6, the utility, in response to the saved state being completed, continues execution by carrying out the appropriate action, which as stated in column 2, lines 6-25 step 3 includes means to display and modify registers and column 3, lines 7-12 and column 1, lines 26-31. Since step 6 is the next sequential step in the synchronized process in Column 9, line 42-65 and includes modifying registers, it

Art Unit: 2183

is inherent the saving of the state of the peripheral must be complete in step 5 before step 6 is started);

Restore the state of the first peripheral, in response to receiving the third signal: (The Test Utility Processor initiates a return to the application program in step 7 (shown in column 9, lines 42-65) via "an assert resume" portion of its software which also indicates the breakpoint has been completed. In the next step (step 8), after the initiation in step 7, the state of the subsystem is restored. (Column 2, lines 66-68 and Column 9, lines 42-65).

Since the target subsystem does carry out the above steps, it is inherent that it is a result of instructions that are stored on a computer readable medium. A target subsystem inherently needs instructions to do any functions, and in order for the instructions to be read by the target subsystem, they are inherently stored on a computer readable medium.

While Fowler teaches that the context/state of the target processor is saved after a breakpoint and before each individual processor is debugged/monitored, Fowler does not teach that the processor which initiates the breakpoint waits for an explicit signal from the target processor which indicates that the target processor has completed saving the state and to continue with the breakpoint handling; a second peripheral coupled to the processor; a digital logic circuit comprising a memory that stores an indicator indicating that the first peripheral is to participate in an execution of the breakpoint but the second peripheral is not to participate, wherein the digital logic circuit is to generate the second signal indicating that the state of the first peripheral has been

Art Unit: 2183

saved, and the digital logic circuit is coupled to the processor, the peripheral, and the second peripheral.

Takase teaches a module that receives a signal that a breakpoint has occurred, (col. 8, line 48 to col. 9, line 19 describe the breakpoint and the signal that indicates a breakpoint has occurred). The state of the module is first saved (col. 9, lines 23-26), and then it outputs a stop notice signal, which indicates the module is stopped and the state has been kept, and then the debugger continues with the execution of the breakpoint. (Column 9, lines 20-42.)

It would have been obvious to one of ordinary skill in the art to have a signal to indicate that the state has finished being saved. Fowler teaches a synchronized breakpoint system and debugging each of the plurality of processors during a breakpoint. Ensuring the system is in a known, stable, and paused state before entering into the debugging operations would cause accurate debugging of the system and prevent altering of data prior to it being saved. This can be ensured using an explicit signal which indicates the processors are in a stopped state and their state is saved, such as that taught by Takase.

Fowler and Takase failed to teach a second peripheral coupled to the processor; a digital logic circuit comprising a memory that stores an indicator indicating that the first peripheral is to participate in an execution of the breakpoint but the second peripheral is not to participate, wherein the digital logic circuit is to generate the second signal indicating that the state of the first peripheral has been saved, and the digital logic circuit is coupled to the processor, the peripheral, and the second peripheral.

However, Yee disclosed a second peripheral coupled to the processor (Yee: Figure 1 element 102)(Element 102 comprises multiple peripherals.);

A digital logic circuit comprising a memory that stores an indicator indicating that the first peripheral is to participate in an execution of the breakpoint but the second peripheral is not to participate (Yee: Figure 2 element 202, column 5 lines 26-53)(The breakpoint control register allows for certain peripherals to receive a breakpoint and disable breakpoints for others. The signals for saving a state, completing the breakpoint, and resuming normal execution are based on the control register.), wherein

The digital logic circuit is to generate the second signal indicating that the state of the first peripheral has been saved (Step 5 in Column 9, line 66 to Column 10, line 22 indicates the peripheral's state has been saved, and afterwards in step 6, the utility, in response to the saved state being completed, continues execution by carrying out the appropriate action, which as stated in column 2, lines 6-25 step 3 includes means to display and modify registers and column 3, lines 7-12 and column 1, lines 26-31. Since step 6 is the next sequential step in the synchronized process in Column 9, line 42-65 and includes modifying registers, it is inherent the saving of the state of the peripheral must be complete in step 5 before step 6 is started)(It would have been obvious to one of ordinary skill in the art at the time of the invention that this signal generation could have moved to the peripheral breakpoint signal because the element is in charge of controlling access of breakpoints to peripheral devices. In addition,



according to "In re Japikse" (181 F.2d 1019, 86 USPQ 70 (CCPA 1950)), shifting the location of parts doesn't give patentability over prior art.), and

The digital logic circuit is coupled to the processor, the peripheral, and the second peripheral (Takase: Figure 2 elements 202 and 102)(The control register is coupled to the peripherals.).

The process of debugging systems has become more complicated as the number of devices connected to processors increase. Testing and debugging becomes a problem of not just testing individual components, but many components in combination. The ability to selectively see certain peripherals and not others assists in more accurately evaluating a plurality of devices at once (Yee: Column 2 lines 49-67 continued to column 3 lines 1-3). One of ordinary skill in the art would have been motivated by allowing more efficient debugging techniques to implement control registers to control which peripherals are being debugged at any given time. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a control register that indicates which peripherals can be debugged at any given time for the advantage of increased efficiency in the debugging process.

17. As per claim 8:

Fowler, Takase, and Yee disclosed the system of claim 7 wherein the first computer-readable medium includes instructions that cause the processor to:

-Resume normal execution in response to receiving a fourth signal: (Step 9, which follows the restoring of the state of the peripheral (step 8), includes releasing a pause-out condition (fourth signal). Following the release of the pause-out condition,



Art Unit: 2183

the utility processor returns to normal operation, step 10. Normally the utility processor has interrupts enabled, but they are disabled for the breakpoint handling, they are re-enabled in step 10, and hence, the utility processor resumes normal operation. Column 9, line 42-65)

-And generate a fifth signal indicating that the processor has resumed normal execution: (Normally the utility processor has interrupts enabled, but they are disabled for the breakpoint handling, they are re-enabled in step 10, and hence, the utility processor resumes normal operation. After the processor (test utility) resumes normal operation, it indicates to the target subsystem (peripheral) to restart execution, which inherently involves a signal being generated and which is also an indication that the processor (test utility) has resumed normal operation. Steps 11-15 in Column 9, line 66 to column 10, line 22, also column 9, lines 23-31)

-And wherein the second computer-readable medium includes instructions that cause the first peripheral to resume normal execution, in response to receiving the fifth signal: (After the processor (test utility) resumes normal operation, it indicates to the target subsystem (peripheral) to restart execution, which inherently involves a signal being generated and which is also an indication that the processor (test utility) has resumed normal operation. Steps 11-15 in Column 9, line 66 to column 10, line 22, also column 9, lines 23-31))

-And wherein the digital logic circuit is configured to generate the fourth signal indicating that the saved state of the first peripheral has been restored: (Figure 8 depicts aspects of the digital logic circuit that generates the fourth signal, pause-out)

Art Unit: 2183

18. As per claim 9:

Fowler, Takase, and Yee disclosed the system of claim 8, further comprising:

-A second processor (Another subsystem with both pause-in/resume-out signals and pause-out/resume-in signals in figure 1); a third computer-readable medium storing instructions that, when applied to the second processor, cause the second processor to:

-Suspend execution and save a state of the second processor, in response to receiving the first signal (pause-in): (Steps 1-4, column 9, line 66 to column 10, line 22, Column 1, lines 18-31 and column 9, lines 14-39))

-Restore the state of the second processor in response to receiving the third signal: (Column 9, lines 36-39, resume-in signal, steps 7)

-And resume normal execution, in response to receiving the fifth signal (pause-in signal): (Steps 11-14)

-And generate the fourth signal (pause-out signal) indicating that the saved state of the second processor has been restored: (Step 8, figure 8 depicts the interface circuitry generating the pause-out signal)

Since the second processor does carry out the above steps, it is inherent that it is a result of instructions that are stored on a computer readable medium. A processor inherently needs instructions to do any functions, and in order for the instructions to be read by the processor, they are inherently stored on a computer readable medium.

While Fowler teaches that the context/state of the target processor is saved after a breakpoint and before each individual processor is debugged/monitored, Fowler does not teach that the processor which initiates the breakpoint waits for an explicit signal

Art Unit: 2183

from the target processor which indicates that the target processor has completed saving the state and to continue with the breakpoint handling.

Takase teaches a module that receives a signal that a breakpoint has occurred, (col. 8, line 48 to col. 9, line 19 describe the breakpoint and the signal that indicates a breakpoint has occurred). The state of the module is first saved (col. 9, lines 23-26), and then it outputs a stop notice signal, which indicates the module is stopped and the state has been kept, and then the debugger continues with the execution of the breakpoint. (Column 9, lines 20-42.)

It would have been obvious to one of ordinary skill in the art to have a signal to indicate that the state has finished being saved. Fowler teaches a synchronized breakpoint system and debugging each of the plurality of processors during a breakpoint. Ensuring the system is in a known, stable, and paused state before entering into the debugging operations would cause accurate debugging of the system and prevent altering of data prior to it being saved. This can be ensured using an explicit signal which indicates the processors are in a stopped state and their state is saved, such as that taught by Takase.

19. As per claim 10:

Fowler, Takase, and Yee disclosed the system of claim 7 including a system on a chip (SOC). (It is inherent that hardware of Fowler, in view of Takase, is constructed on a chip, and therefore, Fowler, in view of Takase, teaches a system (any of a collection of hardware taught in Fowler, in view of Takase) on a chip.)

20. As per claim 11:

Art Unit: 2183

Fowler, Takase, and Yee disclosed the system of claim 7 including a debugging tool (support processor 31) coupled to the system to debug the system (Fowler: Figures 1 and 3, column 4, lines 35-48)

21. As per claim 12:

Fowler, Takase, and Yee disclosed the system of claim 7 wherein the memory comprises a register to store an indicator of whether generation of the second signal is to be based on the state of the first peripheral (Yee: Figure 2 element 202, column 5 lines 26-53)(The breakpoint control register allows for certain peripherals to receive a breakpoint and disable breakpoints for others. The signals for saving a state, completing the breakpoint, and resuming normal execution are based on the control register.).

22. As per claim 13:

Fowler, Takase, and Yee disclosed the system of claim 7 wherein the state identifies a state of internal registers associated with the first peripheral (Yee: Figure 2 element 202, column 5 lines 26-53)(The breakpoint control register allows for certain peripherals to receive a breakpoint and disable breakpoints for others. The signals for saving a state, completing the breakpoint, and resuming normal execution are based on the control register.).

23. As per claim 14:

Fowler, Takase, and Yee disclosed the system of claim 7 wherein the processor operates at a clock rate different than a clock rate of the first peripheral (Fowler: Column 1, lines 18-39).

Art Unit: 2183

24. As per claim 23:

Fowler disclosed an apparatus comprising:

One or more signal lines used to receive signals and to send signals (Fowler: Column 9 lines 42-65)(It's inherent that the signals sent and received for processing the interrupt is done over signal lines.);

A first processor (The test utility system);

A second processor (Target subsystem);

Wherein the first processor is configured to:

Generate a signal indicating that the first processor has encountered the breakpoint (Fowler: Column 1 lines 26-31, column 2 lines 59-67 continued to column 3 lines 1-12, and column 9 lines 42-65 steps 1-4),

Continue execution of the breakpoint in response to receiving a signal on a signal line indicating that the state of a peripheral second processor has been saved (Fowler: Column 9 lines 42-65)(Step 5 saves the state of the target subsystem has been saved.), and

Generate a signal on a signal line indicating that execution of the breakpoint at the first processor has been completed (The Test Utility Processor initiates a return to the application program in step 7 (shown in column 9, lines 42-65) via "an assert resume" portion of its software which also indicates the breakpoint has been completed. In the next step (step 8), after the initiation in step 7, the state of the subsystem is restored. (Column 2, lines 66-68 and Column 9, lines 42-65).

While Fowler teaches that the context/state of the target processor is saved after a breakpoint and before each individual processor is debugged/monitored, Fowler does not teach a third processor; a memory to store an indicator indicating that the second processor is to participate in an execution of a breakpoint but the third processor is not to participate in the execution; that the processor which initiates the breakpoint waits for an explicit signal from the target processor which indicates that the target processor has completed saving the state; and to continue with the breakpoint handling.

However, Takase disclosed a third processor (Takase: Figure 1 element 14)

A module that receives a signal that a breakpoint has occurred, (col. 8, line 48 to col. 9, line 19 describe the breakpoint and the signal that indicates a breakpoint has occurred). The state of the module is first saved (col. 9, lines 23-26), and then it outputs a stop notice signal, which indicates the module is stopped and the state has been kept, and then the debugger continues with the execution of the breakpoint. (Column 9, lines 20-42.) One of ordinary skill in the art would have recognized that using explicit signals, as taught in Takase, ensures the system is in a known, stable, and paused state before entering into the debugging, which, in turn, would cause accurate debugging of the system and prevent altering of data prior to it being saved.

It would have been obvious to one of ordinary skill in the art to have a signal to indicate that the state has finished being saved. Fowler teaches a synchronized breakpoint system and debugging each of the plurality of processors during a breakpoint. One of ordinary skill in the art would have recognized that ensuring the system is in a known, stable, and paused state before entering into the debugging

Art Unit: 2183

operations would cause accurate debugging of the system and prevent altering of data prior to it being saved. This can be ensured using an explicit signal which indicates the processors are in a stopped state and their state is saved, such as that taught by Takase.

Fowler and Takase failed to teach a memory to store an indicator indicating that the second processor is to participate in an execution of a breakpoint but the third processor is not to participate in the execution;

However, Yee disclosed a memory to store an indicator indicating that the second processor is to participate in an execution of a breakpoint but the third processor is not to participate in the execution (Yee: Figure 2 element 202, column 5 lines 26-53)(The breakpoint control register allows for certain peripherals to receive a breakpoint and disable breakpoints for others. The signals for saving a state, completing the breakpoint, and resuming normal execution are based on the control register. It would have been obvious to one of ordinary skill in the art at the time of the invention that this same system could also apply to multiple processors to assist in debugging.).

The process of debugging systems has become more complicated as the number of devices connected to processors increase. Testing and debugging becomes a problem of not just testing individual components, but many components in combination. The ability to selectively see certain peripherals and not others assists in more accurately evaluating a plurality of devices at once (Yee: Column 2 lines 49-67 continued to column 3 lines 1-3). One of ordinary skill in the art would have been

Art Unit: 2183

motivated by allowing more efficient debugging techniques to implement control registers to control which peripherals are being debugged at any given time. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a control register that indicates which peripherals can be debugged at any given time for the advantage of increased efficiency in the debugging process.

25. As per claim 24:

Claim 24 essentially recites the same limitations of claim 2. Therefore, claim 24 is rejected for the same reasons as claim 2.

26. As per claim 25:

Fowler disclosed an apparatus comprising:

One or more signal lines used to receive signals and to send signals (Fowler: Column 9 lines 42-65)(It's inherent that the signals sent and received for processing the interrupt is done over signal lines.);

A first peripheral configured to:

Suspend execution and save a state of the first peripheral (target subsystem) in response to receiving a signal indicating a breakpoint has been encountered (Column 1, lines 26-31, Column 2, lines 59 to column 3, line 12, Steps 1-4 in Column 9, line 42-65); and

Restore the state of the first peripheral, in response to receiving a signal indicating that execution of the breakpoint has been completed (The Test Utility Processor initiates a return to the application program in step 7 (shown in column 9, lines 42-65) via "an assert resume" portion of its software which also indicates



Art Unit: 2183

the breakpoint has been completed. In the next step (step 8), after the initiation in step 7, the state of the subsystem is restored. (Column 2, lines 66-68 and Column 9, lines 42-65);

Fowler failed to teach a second peripheral configured to suspend execution in response to receiving the signal indicating the breakpoint has been encountered.

However, Takase disclosed a second peripheral configured to suspend execution in response to receiving the signal indicating the breakpoint has been encountered (Takase: Column 9 lines 11-26)(The stop signal causes the other modules to suspend execution until the breakpoint is finished executing.).

It would have been obvious to one of ordinary skill in the art to have a signal to indicate that the state has finished being saved. Fowler teaches a synchronized breakpoint system and debugging each of the plurality of processors during a breakpoint. One of ordinary skill in the art would have recognized that ensuring the system is in a known, stable, and paused state before entering into the debugging operations would cause accurate debugging of the system and prevent altering of data prior to it being saved. This can be ensured using an explicit signal which indicates the processors are in a stopped state and their state is saved, such as that taught by Takase.

Fowler and Takase failed to teach a memory to store an indicator indicating that the first peripheral is to participate in the execution of the breakpoint but the second peripheral is not to participate in the execution.

Art Unit: 2183

However, Yee disclosed a memory to store an indicator indicating that the first peripheral is to participate in the execution of the breakpoint but the second peripheral is not to participate in the execution (Yee: Figure 2 element 202, column 5 lines 26-53)(The breakpoint control register allows for certain peripherals to receive a breakpoint and disable breakpoints for others. The signals for saving a state, completing the breakpoint, and resuming normal execution are based on the control register.).

The process of debugging systems has become more complicated as the number of devices connected to processors increase. Testing and debugging becomes a problem of not just testing individual components, but many components in combination. The ability to selectively see certain peripherals and not others assists in more accurately evaluating a plurality of devices at once (Yee: Column 2 lines 49-67 continued to column 3 lines 1-3). One of ordinary skill in the art would have been motivated by allowing more efficient debugging techniques to implement control registers to control which peripherals are being debugged at any given time. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a control register that indicates which peripherals can be debugged at any given time for the advantage of increased efficiency in the debugging process.

27. As per claim 26:

Claim 26 essentially recites the same limitations of claim 3. Therefore, claim 26 is rejected for the same reasons as claim 3.

28. As per claim 30:

Art Unit: 2183

Claim 30 essentially recites the same limitations of claim 27. Therefore, claim 30 is rejected for the same reasons as claim 27.

29. As per claim 32:

Fowler, Takase, and Yee disclosed the system of claim 7 wherein the system comprises a monolithically fabricated chip that includes the processor, the first peripheral, and the second peripheral (Official notice is given that the processor and peripherals could be placed on a single chip.).

30. As per claim 39:

Fowler, Takase, and Yee disclosed the apparatus of claim 23 wherein the apparatus comprises a monolithically fabricated chip that includes the first processor, the second processor, and the third processor (Official notice is given that multiple processors could be placed on a single chip.).

31. As per claim 40:

Claim 40 essentially recites the same limitations of claim 27. Therefore, claim 40 is rejected for the same reasons as claim 27.

32. As per claim 42:

Claim 42 essentially recites the same limitations of claim 32. Therefore, claim 42 is rejected for the same reasons as claim 32.

33. Claims 28, 31, and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fowler (U.S. 4,502,116), in view of Takase (U.S. 6,289,473), further in view of Hicok (U.S. 5,561,761).

Art Unit: 2183

34. As per claim 28:

Fowler and Takase disclosed the method of claim 27.

Fowler and Takase failed to teach wherein saving the state of the input/output device comprises saving the state of a universal asynchronous receiver/transmitter (UART).

However, Hicok disclosed that when debugging a processor using a breakpoint in a multiprocessor environment, it is advantageous to interrogate the elements of the individual processor, including an input/output device (Data Buffer 18 and Data Buffer 20, fig. 1). It would have been obvious to one of ordinary skill in the art to include an input/output device on a processor because, as one of ordinary skill in the art would have recognized, it allows the processor to communicate with external devices and users. The system taught by Hicok allows a non-destructive access, i.e., the state is saved when the CPU is stopped. [Col. 1, lines 55-63 and col. 2, lines 39-59] It would have been obvious to include the ability to debug an input/output device within a processor in a multiprocessor environment to the system of Fowler and Takase because it allows non-destructive interrogation. Therefore, it would have been obvious to include the input/output device and the method of interrogating it, as taught by Hicok, into the system of Fowler and Takase.

However, Fowler, in view of Takase and Hicok, does not specify on the hardware to implement the input and output device.

Art Unit: 2183

Examiner takes Official Notice that it is well known in the art to implement an Input/Output device on a processor as a UART in order to advantageously provide an asynchronous means to communicate with external devices.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to specify the Input/Output devices of Hicok as UARTs since Examiner takes Official Notice asynchronous communication via UARTs is a well known an advantageous method to communicate with external devices.

35. As per claims 31:

Claim 31 essentially recites the same limitations of claim 28. Therefore, claim 31 is rejected for the same reasons as claim 28.

36. As per claim 41:

Claim 41 essentially recites the same limitations of claim 28. Therefore, claim 41 is rejected for the same reasons as claim 28.

### ***Response to Arguments***

37. The arguments presented by Applicant in the response, received on 7/24/2006 are partially considered persuasive.

38. Applicant argues "Takase doesn't teach nor suggest receiving an indication that a first peripheral is to participate in an execution of a breakpoint whereas a second peripheral is not to participate."

This argument is not found to be persuasive for the following reason. The combination of Takase and Fowler reads upon the limitation. Fowler disclosed including

Art Unit: 2183

a first peripheral in the execution of a breakpoint with a processor. Fowler states that the source of the interrupt is known and that the source is involved in the process of executing a breakpoint. Thus, any other peripherals don't participate in the execution of the breakpoint.

39. Applicant argues "Fowler nor Takase disclosed a memory that stores an indicator indicating that the first peripheral is to participate in an execution of the breakpoint but the second peripheral is not to participate" for claims 7 and 25.

This argument is found to be persuasive for the following reason. The examiner agrees that neither reference disclosed this limitation and a new ground of rejection has been given due to the amendment.

40. Applicant argues "Fowler nor Takase disclosed a memory that stores an indicator indicating that the second processor is to participate in an execution of the breakpoint but the third processor is not to participate" for claim 23.

This argument is found to be persuasive for the following reason. The examiner agrees that neither reference disclosed this limitation and a new ground of rejection has been given due to the amendment.

### ***Conclusion***

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

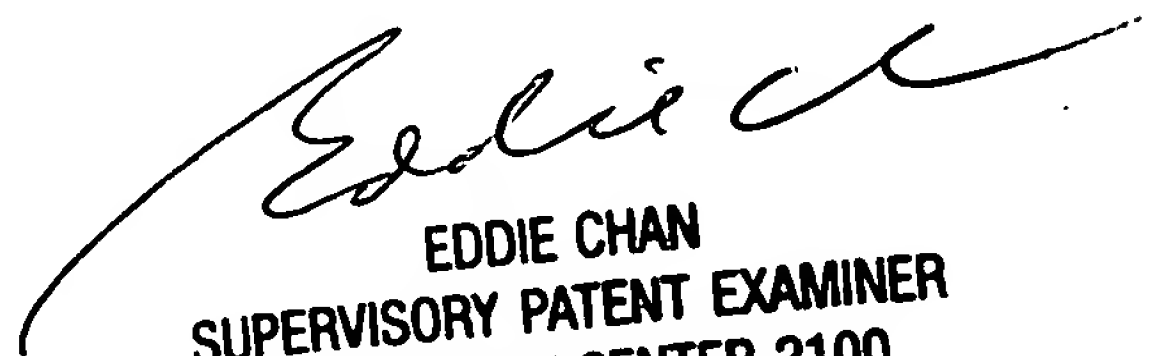
The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob Petranek whose telephone number is 571-272-5988. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jacob Petranek  
Examiner, Art Unit 2183



EDDIE CHAN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100